Simulated VulfenSarah hits landed with first 50 Wolfey as a Testing Set

Janis Corona

11/17/2019

## put in the first 50 samples of Wolfey as the testing set to see the

prediction accuracy of hits landed with VulfenSarah hits landed comparison

Vulfen <- read.csv('SarahWolfEaten\_addedFeatures.csv',   
 sep=',', header=TRUE,   
 na.strings=c('','NA'))

Wolfey <- read.csv('wolfey\_addedFeatures.csv',   
 sep=',', header=TRUE, nrows=50,  
 na.strings=c('','NA'))

Wolfey <- Wolfey[,c(1:7,8:15,48:155)]#omit all X1 landed and x2 received  
Vulfen <- Vulfen[,c(1:7,8:15,48:155)]#omit all X1 landed and x2 received

library(caret)  
library(randomForest)  
library(MASS)  
library(gbm)  
library(dplyr)

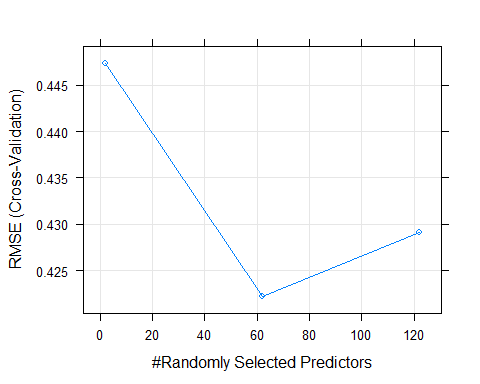
set.seed(189678345)

trainingSet <- Vulfen  
testingSet <- Wolfey

system.time(rfMod <- train(TotLandsX1~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5))

## user system elapsed   
## 35.72 0.20 39.17

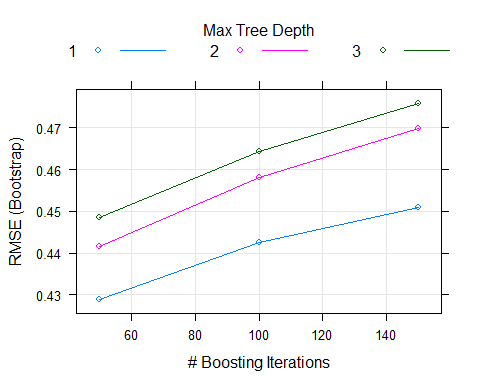
plot(rfMod)



system.time(gbmMod <- train(TotLandsX1~., method='gbm', data=trainingSet, verbose=FALSE ))

## user system elapsed   
## 18.28 0.03 19.76

plot(gbmMod)



predRF <- round(predict(rfMod, testingSet))  
predGbm <- round(predict(gbmMod, testingSet))  
  
predDF <- data.frame(predRF, predGbm, type=testingSet$TotLandsX1)  
predDF

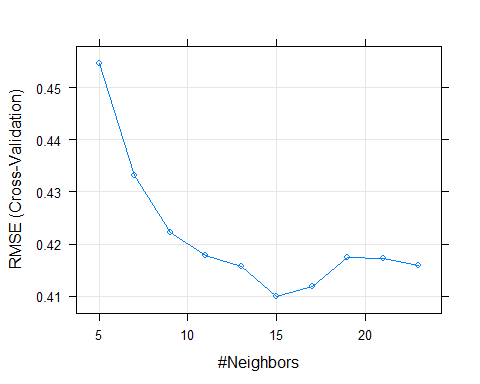
## predRF predGbm type  
## 1 0 0 0  
## 2 0 0 1  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0  
## 7 0 0 0  
## 8 0 0 0  
## 9 0 0 0  
## 10 0 0 0  
## 11 0 0 0  
## 12 0 0 2  
## 13 0 0 0  
## 14 0 0 0  
## 15 0 0 0  
## 16 0 0 0  
## 17 0 0 1  
## 18 0 0 0  
## 19 0 0 0  
## 20 0 0 0  
## 21 0 0 0  
## 22 1 0 0  
## 23 0 0 0  
## 24 1 0 1  
## 25 0 0 0  
## 26 1 0 0  
## 27 0 0 0  
## 28 1 0 0  
## 29 0 0 0  
## 30 1 0 0  
## 31 1 0 0  
## 32 0 0 1  
## 33 0 0 0  
## 34 0 0 0  
## 35 0 0 0  
## 36 1 0 0  
## 37 0 0 0  
## 38 0 0 0  
## 39 1 0 0  
## 40 1 0 0  
## 41 0 0 0  
## 42 0 0 0  
## 43 0 0 0  
## 44 0 0 0  
## 45 0 0 0  
## 46 0 0 0  
## 47 0 0 1  
## 48 0 0 0  
## 49 0 0 1  
## 50 0 0 0

sum <- sum(predRF==testingSet$TotLandsX1)  
length <- length(testingSet$TotLandsX1)  
accuracy\_rfMod <- (sum/length)   
  
sum <- sum(predGbm==testingSet$TotLandsX1)  
accuracy\_Gbm <- (sum/length)

system.time(knnMod <- train(TotLandsX1 ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet))

## user system elapsed   
## 3.38 0.00 5.25

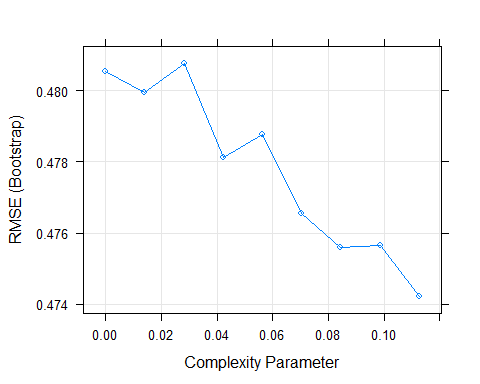
plot(knnMod)



system.time(rpartMod <- train(TotLandsX1~ ., method='rpart', tuneLength=9, data=trainingSet))

## user system elapsed   
## 6.35 0.03 6.82

plot(rpartMod)



system.time(glmMod <- train(TotLandsX1~ .,   
 method='glm', data=trainingSet))

## user system elapsed   
## 2.08 0.00 2.21

predKNN <- round(predict(knnMod, testingSet))  
predRPART <- round(predict(rpartMod, testingSet))  
predGLM <- round(predict(glmMod, testingSet))  
  
df3 <- cbind(predKNN, predRPART, predGLM,testingSet$TotLandsX1)  
colnames(df3)[4] <- 'TrueValue'

length=length(testingSet$TotLandsX1)  
  
sumKNN <- sum(predKNN==testingSet$TotLandsX1)  
sumRPart <- sum(predRPART==testingSet$TotLandsX1)  
sumGLM <- sum(predGLM==testingSet$TotLandsX1)  
  
accuracy\_KNN <- sumKNN/length   
accuracy\_RPART <- sumRPart/length   
accuracy\_GLM <- sumGLM/length   
  
predDF3 <- data.frame(predRF,predGbm,df3)

colnames(predDF3)

## [1] "predRF" "predGbm" "predKNN" "predRPART" "predGLM" "TrueValue"

results <- c(round(accuracy\_rfMod,2),   
 round(accuracy\_Gbm,2),   
 round(accuracy\_KNN,2), round(accuracy\_RPART,2),  
 round(accuracy\_GLM,2),   
 round(100,2))  
  
results <- as.factor(results)  
results <- t(data.frame(results))  
colnames(results) <- colnames(predDF3)  
Results <- rbind(predDF3, results)   
Results

## predRF predGbm predKNN predRPART predGLM TrueValue  
## 1 0 0 0 0 0 0  
## 2 0 0 0 0 0 1  
## 3 0 0 0 0 0 0  
## 4 0 0 0 0 0 0  
## 5 0 0 0 0 0 0  
## 6 0 0 0 0 0 0  
## 7 0 0 0 0 0 0  
## 8 0 0 0 0 0 0  
## 9 0 0 0 0 0 0  
## 10 0 0 0 0 0 0  
## 11 0 0 0 0 0 0  
## 12 0 0 0 0 0 2  
## 13 0 0 0 0 0 0  
## 14 0 0 0 0 0 0  
## 15 0 0 0 0 0 0  
## 16 0 0 0 0 0 0  
## 17 0 0 0 0 0 1  
## 18 0 0 0 0 0 0  
## 19 0 0 1 0 0 0  
## 20 0 0 0 0 1 0  
## 21 0 0 0 0 0 0  
## 22 1 0 0 0 1 0  
## 23 0 0 0 0 0 0  
## 24 1 0 1 0 1 1  
## 25 0 0 0 0 0 0  
## 26 1 0 1 0 1 0  
## 27 0 0 0 0 0 0  
## 28 1 0 1 0 1 0  
## 29 0 0 0 0 0 0  
## 30 1 0 1 0 1 0  
## 31 1 0 1 0 1 0  
## 32 0 0 0 0 1 1  
## 33 0 0 0 0 0 0  
## 34 0 0 0 0 0 0  
## 35 0 0 0 0 0 0  
## 36 1 0 1 0 1 0  
## 37 0 0 1 0 1 0  
## 38 0 0 1 0 1 0  
## 39 1 0 1 0 1 0  
## 40 1 0 1 0 1 0  
## 41 0 0 0 0 1 0  
## 42 0 0 0 0 0 0  
## 43 0 0 0 0 0 0  
## 44 0 0 1 0 1 0  
## 45 0 0 0 0 0 0  
## 46 0 0 0 0 0 0  
## 47 0 0 1 0 1 1  
## 48 0 0 0 0 0 0  
## 49 0 0 1 0 1 1  
## 50 0 0 1 0 1 0  
## results 0.72 0.86 0.68 0.86 0.66 100

bestResults <- Results[,c(2,6)]  
  
# hits predicted for VulfenSarah to land as a simulation of the hits landed by Wolfey's first 50 instances  
  
sum(bestResults[1]>0)# predicted VulfenSarah will land 1 hit

## [1] 1

sum(bestResults[2]>0)# compared to the true value of 12 hits Mazvidal landed

## [1] 8

library(dplyr)  
#BestPredictedHit <- filter(Results,TrueValue == 1 | TrueValue == 2)  
BestPredictedHit <- subset(Results, Results$TrueValue == 1 | Results$TrueValue ==2)  
length=length(BestPredictedHit$TrueValue)  
  
sumRF <- sum(BestPredictedHit$predRF==BestPredictedHit$TrueValue)  
sumGbm <- sum(BestPredictedHit$predGbm==BestPredictedHit$TrueValue)  
sumKNN <- sum(BestPredictedHit$predKNN==BestPredictedHit$TrueValue)  
sumRPart <- sum(BestPredictedHit$predRPART==BestPredictedHit$TrueValue)  
sumGLM <- sum(BestPredictedHit$predGLM==BestPredictedHit$TrueValue)  
  
accuracy\_RF <- round(sumRF/length,2)  
accuracy\_Gbm <- round(sumGbm/length,2)   
accuracy\_KNN <- round(sumKNN/length,2)   
accuracy\_RPART <- round(sumRPart/length,2)   
accuracy\_GLM <- round(sumGLM/length,2)   
Truth <- round(sum(BestPredictedHit$TrueValue==BestPredictedHit$TrueValue)/length,2)  
  
HitAccuracy <- c(accuracy\_RF,accuracy\_Gbm,accuracy\_KNN,accuracy\_RPART,  
 accuracy\_GLM,Truth)  
HitAccuracy <- t(data.frame(as.factor(HitAccuracy)))  
colnames(HitAccuracy) <- colnames(BestPredictedHit)  
BestPredictedHit1 <- rbind(BestPredictedHit,HitAccuracy)  
row.names(BestPredictedHit1)[8] <- 'Accuracy'  
BestPredictedHit1

## predRF predGbm predKNN predRPART predGLM TrueValue  
## 2 0 0 0 0 0 1  
## 12 0 0 0 0 0 2  
## 17 0 0 0 0 0 1  
## 24 1 0 1 0 1 1  
## 32 0 0 0 0 1 1  
## 47 0 0 1 0 1 1  
## 49 0 0 1 0 1 1  
## Accuracy 0.14 0 0.43 0 0.57 1

# KNN and GLM were more accurate in guessing which simulations would produce a hit landed by VulfenSarah out of only 7 hits landed in the testing set, and the accuracy includes the percent correct for hits landed equal to zero for true hits landed equal to zero.

testHits <- testingSet[row.names(BestPredictedHit1)[1:7],]  
Hits <-cbind(BestPredictedHit1[1:7,],testHits)  
Hits

## predRF predGbm predKNN predRPART predGLM TrueValue Round  
## 2 0 0 0 0 0 1 1  
## 12 0 0 0 0 0 2 1  
## 17 0 0 0 0 0 1 1  
## 24 1 0 1 0 1 1 1  
## 32 0 0 0 0 1 1 1  
## 47 0 0 1 0 1 1 1  
## 49 0 0 1 0 1 1 1  
## SecondsIntoRound lastAction SecondsLastRoundAction cmTotHitsR.X1  
## 2 4 1 3 0  
## 12 49 48 1 2  
## 17 67 66 1 2  
## 24 95 94 1 2  
## 32 139 138 1 2  
## 47 235 234 1 2  
## 49 245 238 7 2  
## cmTotHitsL.X1 cmTotHitsM.X1 TotLandsX1 TotMissedX1 TotReceivedX1  
## 2 1 0 1 0 0  
## 12 3 1 2 0 0  
## 17 4 3 1 0 0  
## 24 5 3 1 0 0  
## 32 6 7 1 1 0  
## 47 7 16 1 0 0  
## 49 8 17 1 0 0  
## cmTotHitsR.X2 cmTotHitsL.X2 cmTotHitsM.X2 TotLandsX2 TotMissedX2  
## 2 1 0 0 0 0  
## 12 3 2 0 0 0  
## 17 4 2 0 0 0  
## 24 6 2 0 0 0  
## 32 7 2 1 0 0  
## 47 9 2 5 0 0  
## 49 10 2 5 0 0  
## Crossl.X2 Kneel.X2 Elbowl.X2 Hookl.X2 Jabl.X2 Kickl.X2 upperl.X2  
## 2 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 24 0 0 0 0 0 0 0  
## 32 0 0 0 0 0 0 0  
## 47 0 0 0 0 0 0 0  
## 49 0 0 0 0 0 0 0  
## takedownl.X2 hammerl.X2 Cross2l.X2 Knee2l.X2 Elbow2l.X2 Hook2l.X2  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Jab2l.X2 Kick2l.X2 upper2l.X2 takedown2l.X2 hammer2l.X2 Cross3l.X2  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Knee3l.X2 Elbow3l.X2 Hook3l.X2 Jab3l.X2 Kick3l.X2 upper3l.X2  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## takedown3l.X2 hammer3l.X2 Crossm.X1 Kneem.X1 Elbowm.X1 Hookm.X1 Jabm.X1  
## 2 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 24 0 0 0 0 0 0 0  
## 32 0 0 0 0 0 0 0  
## 47 0 0 0 0 0 0 0  
## 49 0 0 0 0 0 0 0  
## Kickm.X1 upperm.X1 takedownm.X1 hammerm.X1 Cross2m.X1 Knee2m.X1  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Elbow2m.X1 Hook2m.X1 Jab2m.X1 Kick2m.X1 upper2m.X1 takedown2m.X1  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 1 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## hammer2m.X1 Cross3m.X1 Knee3m.X1 Elbow3m.X1 Hook3m.X1 Jab3m.X1  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Kick3m.X1 upper3m.X1 takedown3m.X1 hammer3m.X1 Crossm.X2 Kneem.X2  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Elbowm.X2 Hookm.X2 Jabm.X2 Kickm.X2 upperm.X2 takedownm.X2 hammerm.X2  
## 2 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 24 0 0 0 0 0 0 0  
## 32 0 0 0 0 0 0 0  
## 47 0 0 0 0 0 0 0  
## 49 0 0 0 0 0 0 0  
## Cross2m.X2 Knee2m.X2 Elbow2m.X2 Hook2m.X2 Jab2m.X2 Kick2m.X2 upper2m.X2  
## 2 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 24 0 0 0 0 0 0 0  
## 32 0 0 0 0 0 0 0  
## 47 0 0 0 0 0 0 0  
## 49 0 0 0 0 0 0 0  
## takedown2m.X2 hammer2m.X2 Cross3m.X2 Knee3m.X2 Elbow3m.X2 Hook3m.X2  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Jab3m.X2 Kick3m.X2 upper3m.X2 takedown3m.X2 hammer3m.X2 Crossr.X1  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Kneer.X1 Elbowr.X1 Hookr.X1 Jabr.X1 Kickr.X1 upperr.X1 takedownr.X1  
## 2 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 24 0 0 0 0 0 0 0  
## 32 0 0 0 0 0 0 0  
## 47 0 0 0 0 0 0 0  
## 49 0 0 0 0 0 0 0  
## hammerr.X1 Cross2r.X1 Knee2r.X1 Elbow2r.X1 Hook2r.X1 Jab2r.X1 Kick2r.X1  
## 2 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 24 0 0 0 0 0 0 0  
## 32 0 0 0 0 0 0 0  
## 47 0 0 0 0 0 0 0  
## 49 0 0 0 0 0 0 0  
## upper2r.X1 takedown2r.X1 hammer2r.X1 Cross3r.X1 Knee3r.X1 Elbow3r.X1  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0  
## Hook3r.X1 Jab3r.X1 Kick3r.X1 upper3r.X1 takedown3r.X1 hammer3r.X1  
## 2 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 17 0 0 0 0 0 0  
## 24 0 0 0 0 0 0  
## 32 0 0 0 0 0 0  
## 47 0 0 0 0 0 0  
## 49 0 0 0 0 0 0

## The above table shows the 7 simulations of another fighter in our trained model and the results of those hits that were landed in this testing set against the prediction of a hit landed with those algorithms for machine learning: random forest (rf), global boosted machines (gbm), k nearest neighbors (KNN), recursive partitioning and regression trees (rpart), generalized linear models (glm), and the true testing set value aiming to predict for hits landed.